# SETTING SYSTEM FOR MEMORY ACCESS TIMING

Patent Number: JP63217452
Publication date: 1988-09-09
Inventor(s): SAITO TAKASHI
Applicant(s):: MITSUBISHI ELECTRIC CORP
Requested Patent: JP63217452
Application Number: JP19870051509 19870306
Priority Number(s):
IPC Classification: G06F12/16 ; G11C7/00 ; G11C11/34 ; G11C29/00
EC Classification:
Equivalents:

**Abstract**

PURPOSE:To automatically set the access timing of a memory by changing successively the set value of a register by a program and giving an access to memory based on the set value obtained when the coincidence of comparison is obtained between read data and write data.
CONSTITUTION:An optimum value is set at each of registers 21-24 and a memory (RAM 1) receives an access based on each set value of these registers to write and read the test data. These test write and read data are compared with each other for each access action carried out based on each set value. When the coincidence is obtained between write and read data of the memory (RAM 1), the relevant value is set at the registers 21-24 respectively as the final set value. Then the access timing is decided based on said final set value and the memory (RAM 1) receives accesses in this timing for prescribed data writing and reading operations. In such a way, the access timing of the RAM can be automatically set with no intervention of man power.

HKR178175

HR905_024236

⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　昭63-217452

㊸公開　昭和63年(1988)9月9日

| ⑤Int.Cl.⁴ | 識別記号 | 庁内整理番号 |
| --- | --- | --- |
| G 06 F 12/16 | 310 | A-7737-5B |
| G 11 C 7/00 | 313 | 7341-5B |
| 11/34 | 371 | A-8522-5B |
| 29/00 | 303 | H-7737-5B |

審査請求　未請求　発明の数　1　（全8頁）

⑤発明の名称　メモリアクセスタイミング設定方式

㉑特　願　昭62-51509

㉒出　願　昭62(1987)3月6日

㉒発明者　斎藤　隆　神奈川県鎌倉市上町屋325番地　三菱電機株式会社計算機製作所内

㉑出願人　三菱電機株式会社　東京都千代田区丸の内2丁目2番3号

㉔代理人　弁理士　大岩　増雄　外2名

# 明　細　書

1. 発明の名称

メモリアクセスタイミング設定方式

2. 特許請求の範囲

メモリのアクセスを行う論理回路において、プログラムによって任意に値を設定できるレジスタを設け、このレジスタへの設定値をプログラムによって順々に変えてゆき、メモリへのテストデータの書き込み、読出し、書込みデータと読出しデータとの比較を繰り返して行い、比較の結果が一致した時の設定値をレジスタに設定しておき、この設定値に基づいてメモリのアクセスを行うことを特徴とするメモリアクセスタイミング設定方式。

3. 発明の詳細な説明

（産業上の利用分野）

この発明はデータ処理装置などに備えられるランダムアクセスメモリ（以下RAMと称す）のアクセスタイミングを設定するためのメモリアクセスタイミング設定方式に関するものである。

（従来の技術）

第6図は従来のメモリアクセスタイミング設定方式を採用した論理回路のブロック図である。図において、1はRAM（ダイナミックRAMを用いた場合を例にとる）、2はアドレスマルチプレクサ、3はRAM1とアドレスマルチプレクサ2とを接続するマルチプレクスド・アドレス・バス、4はアドレスマルチプレクサ2に接続されるアドレスバス、5はRAM1に接続されるデータバス、6はメモリ制御リングである。また、7はローアドレスセレクト信号（RAS信号）生成用のフリップフロップ（以下RAS用フリップフロップと称す）、8はカラムアドレスセレクト信号（CAS信号）生成用のフリップフロップ（以下CAS用フリップフロップと称す）、9はカラムセレクト信号（COLS信号）生成用のフリップフロップ（以下COLS用フリップフロップと称す）、10はANDゲート、11，12，13はORゲート、14，15，16はNORゲート、17，18，19，20はメモリ制御リング6のどの出力を使用するかを選択するためのジャンパ線であ

る。なお、説明を簡単にするためＲＡＭ１のリフレッシュ用の論理回路は省略する。

次に動作について説明す。説明の中で「１」は有意もしくはハイレベルを、「０」は非有意もしくはローレベルを意味する。メモリ制御リング６はラインＬ１のメモリアクセスモード信号が「１」になるとイネーブルされて動作可能状態となり、ラインＬ２の基本クロックに同期して出力Ｔ０，Ｔ１，・・・，Ｔｋ，・・・，Ｔℓ，・・・，Ｔｍ，・・・，Ｔｎ，・・・，Ｔｅ－１，Ｔｅがそれぞれ順に「１」になるという形で状態が遷移する。また、メモリアクセスモード信号が「０」になると、メモリ制御リング６の出力Ｔ０～Ｔｅは全て「０」になる。各フリップフロップ７，８，９はラッチした信号を各出力端子１から出力し、そのラッチした信号の反転信号を各出力端子０から出力する。ＲＡＭ１の端子ＲＡＳ，ＣＡＳ，ＷＥにそれぞれ与えられるＲＡＳ信号，ＣＡＳ信号，ＷＥ信号は「１」で有意とする。また、この従来例の場合、ジャンパ線１７，１８，１９，２０の設定は人手により行い、メモリ制御リング６の出力Ｔｋ、Ｔℓ，Ｔｍ，Ｔｎがそれぞれ選択されたものとする。

ここで第４図に示すタイミングチャートを参照してＲＡＭ１への書込み動作を例にとって説明する。メモリアクセスが開始されると、ラインＬ１のメモリアクセスモード信号及びラインＬ３のライトモード信号が「１」になる。この時、アドレスがアドレスバス４に与えられ、アドレスマルチプレクサ２によりローアドレスが選択されてマルチプレクスド・アドレス・バス３上に出力される。また、この時、書込みデータがデータバス５上に与えられる。

このようなローアドレス及び書込みデータが与えられ、メモリ制御リング６は前述したようにラインＬ１のメモリアクセスモード信号が「１」になっているので動作を開始し、出力Ｔ０，Ｔ１，・・・，Ｔｋが順次「１」になるという形で状態が遷移する。そして出力Ｔｋが「１」になると、ジャンパ線１７及びＯＲゲート１１を経由してＲＡＳ用フリップフロップ７の端子Ｄに「１」の出力Ｔｋが与えられ、メモリ制御リング６の出力Ｔｋ＋１が「１」になる時、ＲＡＳ用フリップフロップ７の出力端子１から出力されるＲＡＳ信号が「１」になる。また、この時、ＲＡＳ用フリップフロップ７の出力端子０から出力される反転出力は「０」となり、これによりＮＯＲゲート１４の出力が「１」、ＯＲゲート１１の出力が「１」となって、メモリ制御リング６の状態が進んでもＲＡＳ用フリップフロップ７の出力、すなわちＲＡＳ信号は「１」にホールドされる。また、メモリ制御リング６の「１」の出力がＴℓからＴℓ＋１に遷移した時、前記と同様な動作によりＣＯＬＳ用フリップフロップ９の出力であるＣＯＬＳ信号が「１」になりホールドされる。このＣＯＬＳ信号により、アドレスマチプレクサ２はマルチプレクスド・アドレス・バス３にカラムアドレスを出力し、また、ＡＮＤゲート１０の出力、すなわちＲＡＭ１の端子ＷＥに与えられるＷＥ信号が「１」となり、ＲＡＭ１はライトモードとなる。また、メモリ制御リング６の「１」の出力がＴｍからＴｍ＋１に遷移した時、前記と同様な動作によりＣＡＳ用フリップフロップ８の出力、すなわちＣＡＳ信号は「１」になりホールドされる。以上のようにＲＡＳ信号、ＣＡＳ信号、ＷＥ信号、ＣＯＬＳ信号が全て「１」となって、ＲＡＭ１への書込み条件が全て揃い、データの書込み動作が行われ、メモリ制御リング６の状態が進み、出力Ｔｎ－１が「１」になった時点で書込み動作が完了する。メモリ制御リング６の出力Ｔｎ、すなわちラインＬ４のメモリアクセス完了信号が「１」になり、次に出力Ｔｎ＋１が「１」になろうとするところでラインＬ１のメモリアクセスモード信号及びラインＬ３のライトモード信号が「０」になり、また、ＮＯＲゲート１４，１５，１６及びＯＲゲート１１，１２，１３の出力が「０」になるので、ＲＡＳ信号、ＣＡＳ信号、ＣＯＬＳ信号が「０」になり、ＲＡＭ１への書込み動作が終了する。なお、第４図に示すＴＷは、ＲＡＭ１への制御信号（メモリアクセスモード信号、ライトモ

ード信号、ＲＡＳ信号、ＣＯＬＳ信号、ＣＡＳ信号ＷＥ信号）によるライトモード条件成立期間である。

一方、ＲＡＭ１に対する読出し動作時には、第５図に示すようにライトモード信号及びＷＥ信号は「０」になり、メモリ制御リング６の出力Ｔｎ－１が「１」を出力した終了時点、すなわち出力Ｔｎが「１」になる時点でＲＡＭ１から読み出される出力データが確定しているとし、出力Ｔｎでデータバス５上のデータを取込む。そして、前述の書込み動作時と同様にメモリ制御リング６の出力Ｔｎ＋１が「１」になろうとするところで全ての制御信号が非有意となり、ＲＡＭ１に対するリード動作は完了する。なお、第５図に示すＴＲは、ＲＡＭ１への制御信号によるリードモード条件成立期間である。

〔発明が解決しようとする問題点〕

従来のメモリアクセスタイミング設定方式においては、ＲＡＭへのアクセスタイミングを決定する部分がジャンパ線による設定であったため、そのジャンパ線の設定に人手の介入が必要であった。また、一般にＲＡＭには種々のアクセスタイムのものがあり、使用するＲＡＭの種類を変えたときにはアクセスタイミングを変更するためにジャンパ線の設定をやり直さねばならず、このため設定を誤ってＲＡＭのアクセスが正しく行われなかったり、あるいは高速な動作を行うことができるＲＡＭを使用しているにもかかわらず、低速用のアクセスタイミング設定であるためにＲＡＭ本来の性能を落としてしまうなどの問題点があった。

この発明は上記のような問題点を解消するためになされたもので、ＲＡＭのアクセスタイミングの設定を人手の介入なしに自動的に行い、ＲＡＭの性能を十分に引き出すことができ、信頼性を向上させることができるメモリアクセスタイミング設定方式を提供することを目的とする。

〔問題点を解決するための手段〕

この発明に係るメモリアクセスタイミング設定方式は、メモリ（ＲＡＭ１）のアクセスを行う論理回路において、プログラムによって任意に値を設定できるレジスタ２１，２２，２３，２４を設け、このレジスタ２１，２２，２３，２４への設定値をプログラムによって順々に変えてゆき、メモリ（ＲＡＭ１）へのテストデータの書込み、読出し、書込みデータと読出しデータとの比較を繰り返し、比較の結果が一致した時の設定値をレジスタ２１，２２，２３，２４に設定しておき、この設定値に基づいてメモリ（ＲＡＭ１）のアクセスを行うことを特徴とするものである。

〔作用〕

この発明に係るレジスタ２１，２２，２３，２４にはプログラムによって任意の値が設定され、メモリ（ＲＡＭ１）は設定された各設定値に基づいてアクセスされ、テストデータの書込み、読出しを行う。このテストデータの書込みデータと読出しデータとは各設定値に基づいたアクセス動作毎に比較され、メモリ（ＲＡＭ１）の書込みデータと読出しデータとが一致した時、その値は最終の設定値としてレジスタ２１，２２，２３，２４に設定され、その後はその最終の設定値に基づいてアクセスタイミングが決められ、メモリ（ＲＡＭ１）はそのアクセスタイミングでアクセスされ、所定のデータの書込み、読出し動作を行う。

〔発明の実施例〕

以下この発明の一実施例を図に基づいて説明する。第１図はこの発明の一実施例に係るメモリアクセスタイミング設定方式を採用した論理回路のブロック図である。第１図において、第６図に示す構成要素に対応するものには同一の参照符を付し、その説明を省略する。第１図において、２１はＲＡＳ信号生成タイミングを決めるためのレジスタ（以下ＲＡＳ用レジスタと称す）、２２はＣＯＬＳ信号生成タイミングを決めるためのレジスタ（以下ＣＯＬＳ用レジスタ）、２３はＣＡＳ信号生成タイミングを決めるためのレジスタ（以下ＣＡＳ用レジスタと称す）、２４はメモリアクセス完了信号生成タイミングを決めるためのレジスタ（以下ＣＰＴ用レジスタと称す）、２５，２６，２７，２８はメモリ制御リング６のｅ＋１個ある出力Ｔ０～Ｔｅの内の１個を選択して出力

するセレクタである。

次に動作について説明する。入手できたＲＡＭが例えば５種類あり、そのＲＡＭをアクセスタイムが速い順にＲＡＭ$_1$、ＲＡＭ$_2$、ＲＡＭ$_3$、ＲＡＭ$_4$、ＲＡＭ$_5$とする。その各ＲＡＭに対しては第２図のタイミング図に示すようにＲＡＳ信号、ＣＯＬＳ信号、ＣＡＳ信号、メモリアクセス完了信号を出力すれば各ＲＡＭのアクセスは正しく行える。以下の説明はＲＡＭ$_2$が実装された場合の動作を考える。

プログラムは、ＲＡＳ用レジスタ２１、ＣＯＬＳ用レジスタ２２、ＣＡＳ用レジスタ２３、ＣＰＬＴ用レジスタ２４への設定値、すなわち第２図に対応する設定値ｋ１〜ｋ５．ℓ１〜ℓ５．ｍ１〜ｍ５．ｎ１〜ｎ５をテーブルとして持っている。このプログラムは第３図に示すフローチャートを実行する。すなわち、プログラムが実行され（ステップＳ１）、ポインタはＲＡＭ$_1$を指示し（ステップＳ２）、このポインタの示す内容（この場合第２図に示すＲＡＭ$_1$に対応する設定値ｋ１）をＲＡＳ用レジスタ２１にロードする（ステップＳ３）．次にそのポインタの内容をインクリメントし（ステップＳ４）、ポインタの示す内容（この場合ＲＡＭ$_1$に対応する設定値ℓ１）をＣＯＬＳ用レジスタ２２にロードする（ステップＳ５）次にそのポインタの内容をインクリメントし（ステップＳ６）、ポインタの示す内容（この場合ＲＡＭ$_1$に対応する設定値ｍ１）をＣＡＳ用レジスタ２３にロードする（ステップＳ７）。次にそのポインタの内容をインクリメントし（ステップＳ８）、ポインタの示す内容（この場合ＲＡＭ$_1$に対応する設定値ｎ１）をＣＰＬＴ用レジスタ２４にロードする（ステップＳ９）。次にそのポインタの内容をインクリメントしておき（ステップＳ１０）、テストデータをＲＡＭ$_2$（この場合ＲＡＭ$_2$が実装されているので）に書込みを行う（ステップＳ１１）。この書込みは第４図に示すタイミングで行われる。また、ＲＡＭ$_2$からは第５図に示すタイミングでデータが読出され（ステップＳ１２）、読出しデータと書込みデータとが比較される（ステップＳ１３）。この場合、設定値はＲＡＭ$_1$に対応する設定値ｋ１．ℓ１．ｍ１．ｎ１であり、ＲＡＭ$_2$に対しては制御信号（ＲＡＳ信号、ＣＯＬＳ信号、ＣＡＳ信号、メモリアクセス完了信号）のタイミングが適合しないため、ステップＳ１３での読出しデータと書込みデータとはそのタイミングにおいては等しくならない。従って、ステップＳ１４に移りポインタがエラーか否かを判断し、エラーであるときはエラー報告し（ステップＳ１５）、エラーでないときはステップＳ３に戻る。

このステップＳ３に戻ったときのポインタの示す内容は実装されているＲＡＭ$_2$に対応する設定値ｋ２になっており、この設定値ｋ２がＲＡＳ用レジスタ２１にロードされる。その後は、前述と同様な処理を行い（ステップＳ４〜Ｓ１０）、設定値ℓ２がＣＯＬＳ用レジスタ２２に、設定値ｍ２がＣＡＳ用レジスタ２３に、設定値ｎ２がＣＰＬＴ用レジスタ２４にそれぞれロードされ、テストデータのＲＡＭ$_2$への書込み（ステップＳ１１）、ＲＡＭ$_2$からのデータ読出し（ステップＳ１２）を行い、読出しデータと書込みデータとが比較される（ステップＳ１３）。この場合に、設定値ｋ２．ℓ２．ｍ２．ｎ２がＲＡＭ$_2$に対応しているのでＲＡＭ$_2$は所定のタイミングでアクセスされ、従って、読出しデータと書込みデータとが等しくなり、ステップＳ１６に移り設定値ｋ２．ℓ２．ｍ２．ｎ２が各レジスタ２１．２２．２３．２４への最終の設定値として設定され、セレクタ２５．２６．２７．２８によってメモリ制御リング６の出力Ｔｋ２＋１が「１」の時にＲＡＳ信号が、出力Ｔℓ２＋１が「１」の時にＣＯＬＳ信号が、出力Ｔｍ２＋１が「１」の時にＣＡＳ信号が、出力Ｔｎ２が「１」の時にメモリアクセス完了信号がそれぞれ「１」になり、また、出力Ｔｎ２＋１が「１」の時にＲＡＳ信号、ＣＯＬＳ信号、ＣＡＳ信号、メモリアクセス完了信号がそれぞれ「０」になるというＲＡＭアクセスタイミングが設定され、所定のデータの書込み、読出し動作が行われる。

なお、このフローチャートの説明はRAM$_0$が実装されている場合について述べたが、実装されているRAMが、RAM$_1$，RAM$_2$，RAM$_3$，RAM$_4$のときはステップS3～S13の処理は一回，3回，4回，5回それぞれ行われ、アクセスタイミングが設定される。

上記実施例によれば、プログラムによってアクセスタイミングの変更が可能なことから、RAMのアクセスタイミングのマージン試験を簡単に行うこともできる。また、アクセスタイムの一番遅い種類のRAMのタイミングになってしまうが、アクセスタイムの異なるRAMが混在して論理回路に実装されたとしても一応RAMアクセスが正常に行うことができる。さらに、高速な計算機において、マイクロプログラムにより主記憶のカード単位あるいはバンク単位に別々にRAMアクセスタイミングを設定するようにすれば、カード単位あるいはバンク単位でRAMの種類が異なっても、そのRAMの性能に合ったタイミングでアクセスが行われ、RAMの種類の混在による性能低下は防ぐことができる。また、設定値の決定を行うプログラムにおいて、よりきめ細かく設定値を変えて最適値を選ぶことによりRAMの製造元毎に異なる微妙なタイミングの違いを吸収するように設定値を決めるのではなく、実装されたRAMに最も適した、あるいはRAMの性能を十分に引き出せるアクセスタイミングを設定することができるのる。また、プログラムにおいて全RAMの全アドレスに対してアクセスタイミング設定のチェックを行うようにすれば、例えば別の種類（アクセスタイム）のRAMが混在していた場合に、どのRAMが異常であるかを判別することもできる。

なお、上記実施例においてはRAMとしてダイナミックRAMを用いた場合を示したが、スタティックRAMを用いた場合にはRAS信号及びCAS信号の代わりにチップセレクト信号（CS信号）及びアウトプットイネーブル信号（OE信号）を制御するようにすればよい。また、上記実施例ではメモリ制御リングを用いてRAMアクセスタイミングを制御したが、データロードの可能なカウンタと、カウンタに初期値としてロードする値を設定するレジスタの組合わせをプログラム制御したい信号毎に設けることによっても本発明の方式は実現できる。

〔発明の効果〕

以上のように本発明によれば、プログラムによって任意に値を設定できるレジスタを設け、このレジスタへの設定値をプログラムによって順々に変えてゆき、メモリへのテストデータの書込み、読出し、書込みデータと読出しデータとの比較を繰り返し、比較の結果が一致した時の設定値をレジスタに設定しておき、この設定値に基づいてメモリのアクセスを行うようにしたのでメモリのアクセスタイミングの設定を人手の介入なしに自動的に行うことができ、これによりアクセスタイミングの設定ミスによりメモリの性能の低下やアクセスが正常に行えないということがなくなり、従って、メモリの性能を十分に引き出すことができ、信頼性を向上させるという効果が得られ、また、従来のようにジャンパ線によるアクセスタイミング設定のための人手の介入が不必要となり、これにより動作試験費あるいは人件費が削減でき、より安価なデータ処理装置などを提供することができるという効果が得られる。

4. 図面の簡単な説明

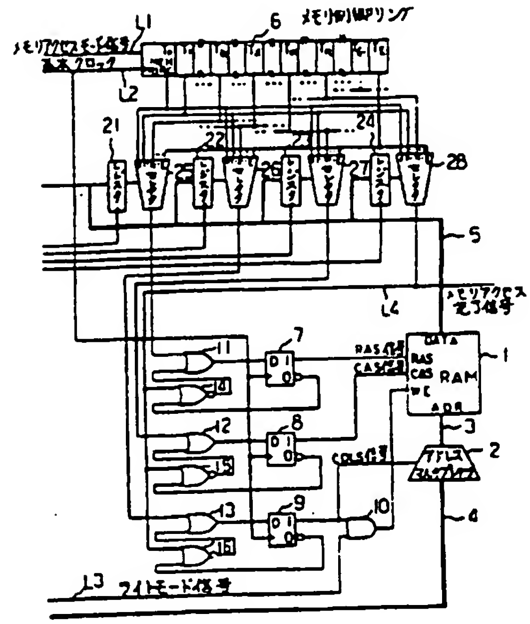
第1図はこの発明の一実施例に係るメモリアクセスタイミング設定方式を採用した論理回路のブロック図、第2図はこの実施例におけるアクセスタイミングと設定値との関係を示すタイミング図、第3図はこの実施例の動作を説明するためのフローチャート、第4図は従来例及びこの実施例のRAMに対する書込み動作を説明するためのタイミングチャート、第5図は従来例及びこの実施例のRAMに対する読出し動作を説明するためのタイミングチャート、第6図は従来のメモリアクセスタイミング設定方式を採用した論理回路のブロック図である。

1・・・RAM（メモリ）、2・・・アドレスマルチプレクサ、6・・・メモリ制御リング、7・・・RAS用フリップフロップ、8・・・CA

S用フリップフロップ、9・・・COLS用フリップフロップ、10・・・ANDゲート、11、12、13・・・ORゲート、14、15、16・・・NORゲート、21・・・RAS用レジスタ、22・・・COLS用レジスタ、23・・・CAS用レジスタ、24・・・CPLT用レジスタ、25、26、27、28・・・セレクタ。

代理人　大　岩　増　雄（ほか2名）

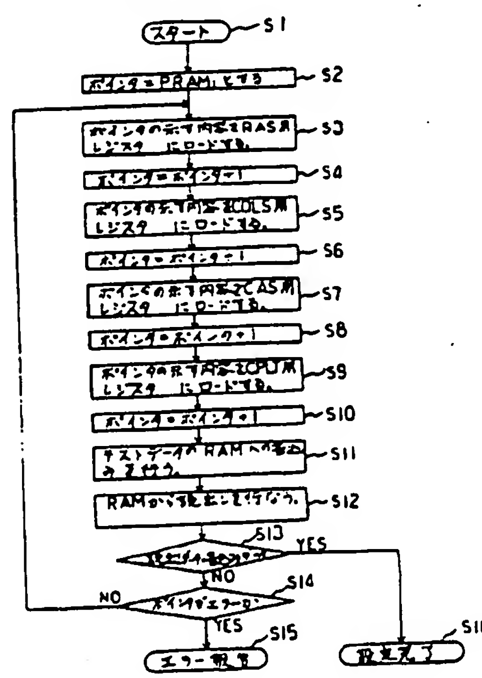
第1図




第2図

| RAMの種類 | RAS用タイミング | COLS用タイミング | CAS用タイミング | [illegible] | タイミング[illegible]レジスタ21 | [illegible]レジスタ22 | [illegible]レジスタ23 | [illegible]レジスタ24 |
|---|---|---|---|---|---|---|---|---|
| RAM1 | Tk1+1 | Te1+1 | Tm1+1 | Tn1 | k1 | ℓ1 | m1 | n1 |
| RAM2 | Tk2+1 | Te2+1 | Tm2+1 | Tn2 | k2 | ℓ2 | m2 | n2 |
| RAM3 | Tk3+1 | Te3+1 | Tm3+1 | Tn3 | k3 | ℓ3 | m3 | n3 |
| RAM4 | Tk4+1 | Te4+1 | Tm4+1 | Tn4 | k4 | ℓ4 | m4 | n4 |
| RAM5 | Tk5+1 | Te5+1 | Tm5+1 | Tn5 | k5 | ℓ5 | m5 | n5 |

第3図

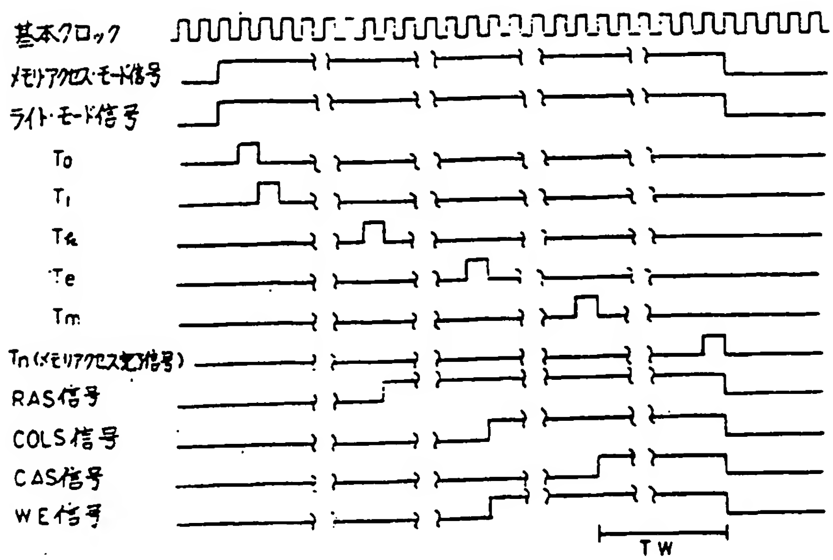




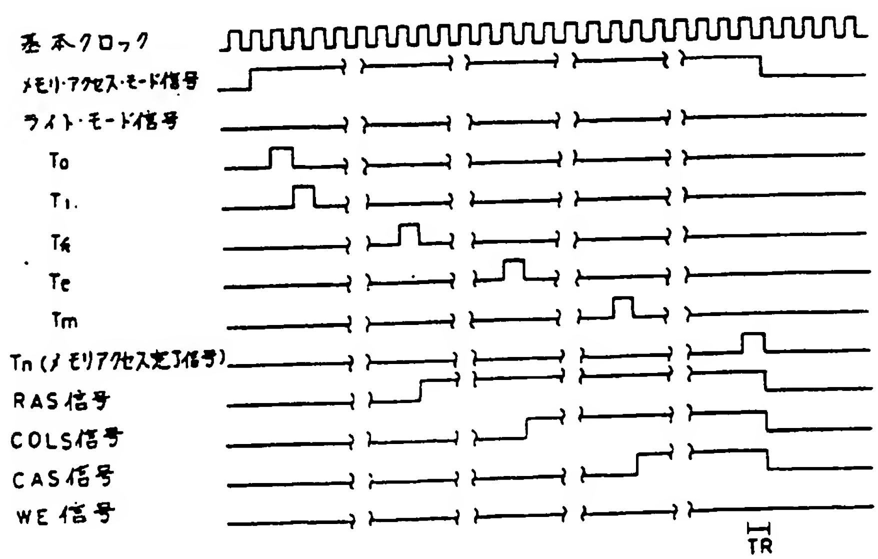

HKR178181

HR905_024242

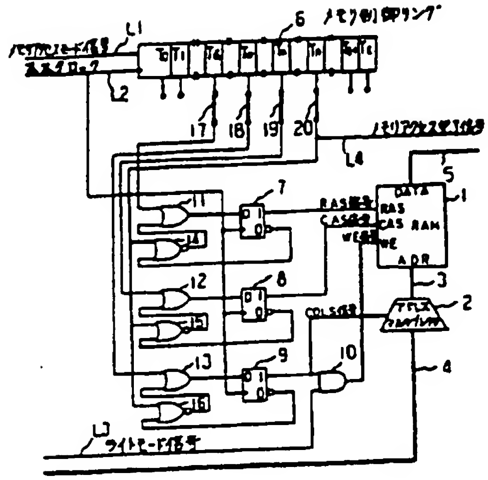
第4図


基本クロック
メモリアクセス・モード信号
ライト・モード信号
T0
T1
Ta
Te
Tm
Tn (メモリアクセス完了信号)
RAS信号
COLS信号
CAS信号
WE信号
TW

第5図


基本クロック
メモリ・アクセス・モード信号
ライト・モード信号
T0
T1
Ta
Te
Tm
Tn (メモリアクセス完了信号)
RAS信号
COLS信号
CAS信号
WE信号
TR

第6図


メモリ制御リング
6
L1
L2
17
18
19
20
L4
5
DATA
RAS
CAS
RAM
WE
ADR
1
3
2
4
7
8
9
10
11
12
13
14
15
16
L3
ライトモード信号

(19) Japan Patent Office (JP)
(11) Patent Application Disclosure
(12) Unexamined Patent Application Publication (A) S63-217452

| (51) Int. Cl.[4] | Identification No. | File No. |
|---|---|---|
| G 06 F 12/16 | 310 | A-7737-5B |
| G 11 C 7/00 | 313 | 7341-5B |
| 11/34 | 371 | A-8522-5B |
| 29/00 | 303 | H-7737-5B |

(43) Disclosure Date: September 9, 1988

No examination requested
Number of inventions: 1 (Total of 8 pages)

(54) Name of Invention: Method for Setting Memory Access Timing

(21) Patent Application S62-51509
(22) Application Date: March 6, 1987

(72) Inventor: Takashi Saito
Mitsubishi Electric Company, Ltd., Computer Manufacturing Facilities
325 Uemachiya, Kamakura City, Kanagawa Prefecture
(71) Applicant: Mitsubishi Electric Company, Ltd.
2-2-3 Marunouchi, Chiyoda-ku, Tokyo
(74) Representative: Masao Oiwa, Patent Attorney, and two others

Details

1. Name of Invention
Method for Setting Memory Access Timing

2. Patent Claims
A method for setting memory access timing in a logical circuit that accesses memory, characterized by the provision of a register for which a variety of values can be set by a program, by repetitively having the value set in said register be changed sequentially by the program and performing test writes to the memory and test reads from the memory where the data that is written is compared to the data that is read, by setting to the register the setting that was in effect when the results of the comparison matched, and by accessing the memory based on said setting.

3. Detailed Explanation of the Invention

[Area of Application in Industry]

This invention pertains to a method for setting the memory access timing in order to set the access timing for random access memory (hereinafter abbreviated "RAM") that is equipped in, for example, data processing devices.

[Prior Art]

Figure 6 shows a block diagram of the logic circuits that use the conventional memory access timing set method. In the figure, 1 is RAM (using the example where a dynamic RAM is used), 2 is an address multiplexer, 3 is a multiplexed address bus connecting the RAM 1 with the address multiplexer 2, 4 is an address bus connected to the address multiplexer 2, 5 is a data bus connected to the RAM 1, and 6 is a memory control ring. In addition, 7 is a flipflop for generating the row address select signal (RAS signal), hereinafter abbreviated "RAS flipflop," 8 is a flipflop for generating the column address select signal (CAS signal), hereinafter abbreviated "CAS flipflop," 9 is a flipflop for generating the column select signal (COLS signal), hereinafter abbreviated "COLS flipflop," 10 is an AND gate, 11, 12, and 13 are OR gates, 14, 15, and 16 are NOR gates, and 17, 18, 19, and 20 are jumper lines for selecting the output from the memory control ring 6 that is to be used. Additionally, for simplicity in the explanation, the logic circuits for refreshing the RAM 1 are not shown.

The operation of this method is described below. In this explanation, "1" indicates either the active level or the high logic level, while "0" indicates the inactive level or the low logic level. The memory control ring 6 is enabled and placed in an operational state when the memory access mode signal of line L1 goes to "1," and, synchronized to the master clock on line L2, the outputs T0, T1, . . . Tk, . . ., Tl, . . ., Tm, . . ., Tn, . . ., Te-1, Te are sequentially set to "1" in the state transitions. When the memory access mode signal is "0," all outputs T0 to Te from the memory control ring 6 go to "0." The respective flipflops 7, 8, and 9 each output their latched signals from the output terminal 1 on each, and output the inverse of the latched signal on output terminal 0 of each. The RAS signal, CAS signal, and WE signal applied, respectively, to the RAS, CAS, and WE terminals of RAM 1 are each active at "1." In addition, in this conventional example, jumper lines 17, 18, 19, and 20 are set by hand, selecting, respectively, outputs Tk, Tl, Tm, and Tn of the memory control ring 6.

Below will be explained an example of an operation to write to the RAM 1, referencing the timing chart shown in Figure 4. When the memory access commences, both the memory access mode signal on Line L1 and the write mode signal on Line 3 both go to "1." At this time, the address is applied to the address bus 4, the row address is selected by the address multiplexer 2 and is output on the multiplexed address bus 3. At this time the write data is applied to data bus 5.

In this way, the row address and write data are applied, and, as described above, the memory access mode signal of line L1 is at "1," so the memory control ring 6 commences operations, and there are state transitions so that outputs T0, T1, . . . Tk sequentially go to "1." When Tk goes to "1," the "1" output Tk is applied to terminal D of the RAS flipflop 7 through the jumper line 17 and OR gate 11, and when output Tk + 1 of memory control ring 6 goes to "1," the RAS signal that is output from output terminal 1 of the RAS flipflop 7 goes to "1." In addition, at this time the inverted signal that is output from the output terminal 0 of the RAS flipflop 7 goes to "0," causing the output of the NOR gate 14 to go to "1," and the output of the OR gate 11 to go to "1," causing the output of the RAS flipflop 7, or in other words the RAS signal, to be held at "1" even if the memory control ring 6 status advances. When the "1" output of the memory control ring 6 transitions from Tl to Tl + 1, the same operation as described above causes the COLS signal, which is the output of the COLS flipflop 9, to be held at "1." This COLS signal causes the address multiplexer 2 to output the column address to the multiplexed address bus 3, and the output of the AND gate 10, or in other words the WE signal that is applied to the RAM 1 terminal WE, goes to "1," placing RAM 1 in write mode. When the "1" output of the memory control ring 6 transitions from Tm to Tm + 1, a operation similar to what was described above causes the output of the CAS flipflop 8, or in other words the CAS signal, to be held at "1." As described above, the RAS signal, the CAS signal, the WE signal, and the COLS signal all go to "1," putting all conditions in place to write to the RAM 1; hence the data write operation is performed, the status of the memory control ring 6 advances, and the write operation is concluded at the point in time where the output Tn - 1 goes to "1." When the output Tn of the memory control ring 6, or in other words the memory access complete signal on line 4, goes to "1" followed by the output Tn + 1 going to "1," the memory access mode signal on line L1 and the write mode signal on line L3 both go to "0," causing the outputs of the NOR gates 14, 15, and 16, along with the outputs of the OR gates 11, 12, and 13 to go to "0"; consequently, the RAS signal, the CAS signal, and the COLS signal all go to "0," completing the operation for writing to the RAM 1. Note that TW shown in Figure 4 is the period over which the write mode conditions are fulfilled by the control signals to the RAM 1 (i.e., the memory access mode signal, the write mode signal, the RAS signal, the COLS signal, the CAS signal, and the WE signal).

On the other hand, in the operations to read from the RAM 1, as shown in Figure 5, the write mode signal and the WE signal go to "0," and at the point in time when the output Tn - 1 of the memory control ring 6 ceases to output "1," or in other words, at the point in time when the output Tn goes to "1," the output data that is read from the RAM 1 is assumed to be set, and with the output Tn, the data on the data bus 5 is accepted. At this time, when, in operations similar to the write operations described above the output Tn + 1 of the memory control ring 6 is to go to "1," all control signals become inactive and the operations to read from the

RAM 1 are terminated. Note that the TR shown in Figure 5 is the period of time over which the control signals to the RAM 1 fulfill the read mode conditions.

[Problems Solved by this Invention]

In the conventional method for setting the memory access timing, the part that sets the RAM access timing is set by jumper lines, and thus it requires a manual intervention to set the jumper lines. Additionally, generally RAMs have a variety of different access times, and when the type of RAM that is used is changed it is necessary to change the settings of the jumpers in order to change the access timing, and, as a result, the RAM cannot be accessed correctly if the setting is incorrect or there may also be the problem that, even if RAM that can operate at high speeds is used, the actual performance of the RAM will not be good if the access timing used is for low speed RAM.

This invention was created in order to solve the types of problems described above, and its objective is to provide a method of setting the memory access timing that automatically sets the access timing without any manual intervention, making it possible to exploit the full capabilities of the RAM and to improve reliability.

[Method by Which the Problems are Solved]

The method of setting the memory access timing in this invention is characterized by the logic circuits that access the memory (RAM 1) being equipped with registers 21, 22, 23, and 24 that can be set to a variety of values by a program, where the values that are set to these registers 21, 22, 23, and 24 are repetitively changed sequentially by the program at which time test data is written to and read from the memory (RAM 1) and comparisons are made between the write data and the read data where the values that were set when the results of the comparison indicates a match are set to registers 21, 22, 23, and 24, so that the access to the memory (RAM 1) is performed based on these settings.

[Operation]

The registers 21, 22, 23, and 24 in this invention are set to any given value by the program, and the memory (RAM 1) is accessed based on the various settings that have been set, at which time test data is written to the memory and read from the memory. The data written as this test data, and the data that is read, are compared to each other for each of the access operations that are based on the respective settings, and when the data that is written to the memory (RAM 1) matches the data that is read from the memory, then the settings are set as the final settings in the registers 21, 22, 23, and 24, and after that time the access timing is determined based on these final settings and the memory (RAM 1) is accessed with that access timing when the specific data write and data read operations are performed.

[Example of Embodiment]

An Example of Embodiment of this invention is explained below based on the figures. Figure 1 is a block diagram of logic circuits that use the method for setting the memory access timing in this Embodiment of the invention. In Figure 1 the same symbols are used as corresponding to the structural elements shown in Figure 6, so the explanations are omitted here. In Figure 1, 21 is the register for determining the timing with which the RAS signal is produced (hereinafter termed the "RAS register"), 22 is the register for determining the timing with which the COLS signal is produced (hereinafter termed the "COLS register"), 23 is the register for determining the timing with which the CAS signal is produced (hereinafter termed the "CAS register"), 24 is the register for determining the timing with which the memory access complete signal will be produced (hereinafter termed the "CPLT register"), 25, 26, 27, and 28 are the selectors that select one output from output T0 to Te of the e + 1 registers in memory control ring 6.

Next the operation will be explained. Let us assume that there are five different types of RAM that can be obtained, and, the access timing on these types of RAM, from fastest to slowest, are $RAM_1$, $RAM_2$, $RAM_3$ $RAM_4$, and $RAM_5$. The respective RAMs can be accessed correctly by outputting the RAS signals, COLS signals, CAS signals, and memory access complete signals shown in the timing diagram of Figure 2. The explanation described below considers the operations when $RAM_2$ is installed.

The table has the settings for the RAS register 21, the COLS register 22, the CAS register 23, and the CPLT register 24, or in other words, the settings for k1 to k5, l1 to l5, m1 to m5, and n1 to n5 in Figure 2, are stored as a table. This program executes the flow chart shown in Figure 3. In other words, the program is executed (Step S1), the pointer indicates $RAM_1$ (Step S2), the information indicated by the pointer (in this case, the settings k1 corresponding to $RAM_1$ shown in Figure 2) are loaded into RAS register 21 (Step S3), the pointer is then incremented (Step S4), the information indicated by the pointer (in this case, the setting l1 corresponding to $RAM_1$) is loaded into the COLS register 22 (Step S5), the pointer is incremented (Step S6), the information indicated by the pointer (in this case, the setting m1 corresponding to $RAM_1$) is loaded into the CAS register 23 (Step S7), the pointer is incremented (Step S8), the information indicated by the pointer (in this case the setting n1 corresponding to $RAM_1$) is loaded into the CPLT register 24 (Step S9), the pointer is incremented (Step S10), the test data is written into the $RAM_2$ (because in this case it is $RAM_2$ that is installed) (Step S11), and the write operation is performed with the timing shown in Figure 4. Then the data is read from the $RAM_2$ with the timing shown in Figure 5 (Step S12), and the data that was read is compared to the data that was written (Step S13). In this case, the settings are the settings k1, l1, m1, and n1 that correspond to $RAM_1$. These settings do not match the timing for the control signals (the RAS signal, the COLS signal, the CAS signal, and the memory access complete signal) for $RAM_2$, so the comparison in Step 13 of the data that was read and the data that

was written does not indicate a match with this timing. As a result, the program continues to Step S14, and a check is made for a pointer error. If there is an error then an error report is made (Step S15), and if there is no error, then the program returns to Step S3.

The information indicated by the pointer when the program returns to Step S3 is the setting k2 that corresponds to the $RAM_2$ that is installed, and this setting k2 is loaded into the RAS register 21. After that, the same process that is described above is performed (Steps S4 through S10) and the setting l2 is loaded into the COLS register 22, the setting m2 is loaded into the CAS register 23, the setting n2 is loaded into the CPLT register 24, the test data is written to the $RAM_2$ (Step S11) the data is read from the $RAM_2$ (Step S12), and the data that was written is compared to the data that was read (Step 13). In this case, the RAM access timing is set so that, when the operations for writing and reading the specified data are performed, the settings k2, l2, m2, and n2 correspond to $RAM_2$, and thus $RAM_2$ is accessed with the appropriate timing and the data that was read matches the data that was written so the program continues to Step 16 and the settings k2, g2, m2, and n2 are set into registers 21, 22, 23, and 24 as the final settings, and selectors 25, 26, 27, and 28 cause the RAS signal to be "1" when the output Tk2 + 1 of the memory control ring 6 is "1," the COLS signal to be "1" when the output Tl2 + 1 is "1," the CAS signal to be "1" when the output Tm2 + 1 is "1," and the memory access complete signal to be "1" when the output Tn2 is "1." In addition, when the output Tn2 + 1 is "1" the RAS signal, the COLS signal, the CAS signal and the memory access complete signal all go to "0."

While the explanation of the flow chart was based on the assumption that RAM 2 was installed, if $RAM_1$, $RAM_3$, $RAM_4$, or $RAM_5$ were installed instead, the processes in Steps 3 through 13 would be performed once, three times, four times, or five times, respectively, to set the access timing.

Because in the Example of Embodiment described above, it is possible to change the access timing using a program, it is easy to perform RAM access timing margin tests. In addition, although the timing will be that for the type of RAM with the slowest access time, even if a mixture of RAMs with different access times are installed in the logic circuits, the RAM can still be accessed correctly. In addition, if in high-speed computers, the RAM access timing is set individually by the card unit or the bank unit of main memory, then even if the type of RAM is different on different card units or bank units, the timing can be performed to match the capability of the RAM, making it possible to prevent any impediments to performance by mixing types of RAM. Additionally, in the program that determines the settings, it is possible to set the access timing that is optimized for the RAM that is installed and that is able to fully exploit the capabilities of the RAM through selecting the optimal values through changing the settings in even finer increments, rather than determining the settings in such a way as to compensate for the minor timing differences between the various RAM manufacturing locations. If in the program access timing setting checks are

performed for all addresses of all RAM, then it is possible to identify the RAM that has errors even if different types of RAM (with different access times) are mixed.

Furthermore, in the Example of Embodiment described above, dynamic RAM was used as the example, when static RAM is used then the chip select (CS) signal and the output enable (OE) signal can be controlled instead of the RAS signal and the CAS signal. Although in the Example of Embodiment above a memory control ring was used to control the RAM access timing, the method of this invention can also be performed by establishing for each signal to be controlled by the program a combination of a counter into which data can be loaded and a register that sets the value that is loaded into the counter as the initial value.

[Effects of the Invention]

Using the invention described above, it is possible to set the memory access timing automatically without a manual intervention because a register is provided wherein a variety of different values can be set by the program where the values that are set into this register are repetitively changed sequentially, test data is written to the memory and then read from the memory, and the data that was written is compared to the data that is read and the values that were set when the results of the comparison indicate a match are set to the register so that the memory is accessed based on those settings, it is able to prevent any disruptions to memory performance or situations where access cannot be performed normally due to incorrect settings in the access timing, making it possible to fully exploit the capabilities of the memory, and thus possible to obtain the effect of increased reliability; in addition, it is no longer necessary to have a manual intervention in order to set the access timing using jumper lines as it has been conventionally, thus making it possible to reduce operating test expenses and reduce labor expenses, and making it possible to provide data processing equipment less expensively.

4. Simple Explanation of Figures

Figure 1 is a block diagram of the logic circuits that use the method for setting the memory access timing in the Example of Embodiment of this invention.
Figure 2 is a timing diagram showing the relationship between the settings and the access timing in this Example of Embodiment.
Figure 3 is a flow chart used for explaining the operation of the Example of Embodiment.
Figure 4 is a timing chart for explaining a conventional example and explaining the operations for writing to the RAM in the Example of Embodiment thereof.
Figure 5 is a timing chart for explaining the conventional example and for explaining the operations for reading from the RAM in an example thereof.
Figure 6 is a block diagram of the logic circuits that use the conventional method of setting the memory access timing.

1: RAM (memory)
2: Address multiplexer
6: Memory control ring
7: RAS flipflop
8: CAS flipflop
9: COLS flipflop
10: AND gate
11, 12, 13: OR gates
14, 15, 16: NOR gates
21: RAS register
22: COLS register
23: CAS register
24: CPLT register
25, 26, 27, 28: Selectors

Representative: Masao Oiwa, and two others

Figure 1
[L1] Memory access mode signal
[L2] Fundamental clock
[6] Memory control ring
[21] Register
[25] Selector
[22] Register
[26] Selector
[23] Register
[27] Selector
[24] Register
[28] Selector
[L4] Memory access complete signal
[Under 14] RAS signal
CAS signal
[2] Address multiplexer
[Above 10] COLS signal
[L3] Write mode signal

Figure 2
[INSERT TABLE]

| Type of RAM | RAS signal timing | COLS signal timing | CAS signal timing | Memory access complete signal timing | Settings for the registers for generating the timing | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | Registe | Registe | Registe | Registe |
| | | | | | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | r 21 | r 22 | r 23 | r 24 |
| | | | | | | | | |
| [see source for English] | | | | | | | | |

Figure 3

S1 Start
S2 Set pointer = PRAM 1
S3 Load the information indicated by the pointer into the RAS register
S4 Pointer = pointer + 1
S5 Load the information indicated by the pointer into the COLS register
S6 Pointer = pointer + 1
S7 Load the information indicated by the pointer into the CAS register
S8 Pointer = pointer + 1
S9 Load the information indicated by the pointer into the CPLT register
S10 Pointer = pointer + 1
S11 Write test data to the RAM
S12 Read test data from the RAM
S13 Compare the read data to the write data
S14 Is there a pointer error?
S15 Error report
S16 Settings complete

Figure 4

Fundamental clock
Memory access mode signal
Write mode signal
T0
Tl
Tk
Te
Tm
Tn (Memory access complete signal)
RAS signal
COLS signal
CAS signal
WE signal

Figure 5

Fundamental clock
Memory access mode signal
Write mode signal
T0
Tl
Tk
Te
Tm
Tn (Memory access complete signal)

RAS signal
COLS signal
CAS signal
WE signal

Figure 6
[L1] Memory access mode signal
[L2] Fundamental clock
[6] Memory control ring
[L4] Memory access complete signal
[Under L4] RAS signal
CAS signal
WE signal
[2] Address multiplexer
[Above 10] COLS signal
[L3] Write mode signal